US 20060159028A1

(19) **United States**

(12) **Patent Application Publication**

Curran-Gray et al.

(10) **Pub. No.: US 2006/0159028 A1**

(43) **Pub. Date: Jul. 20, 2006**

(54) **MONITORING SYSTEM, METHOD OF SAMPLING DATAGRAMS, AND APPARATUS THEREFOR**

(76) Inventors: **Martin Curran-Gray**, Fife (GB); **Andrew Robert Lehane**, Milnathort (GB)

Correspondence Address:
Paul D. Greeley, Esq.
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
10th Floor
One Landmark Square
Stamford, CT 06901-2682 (US)

(21) Appl. No.: **11/273,128**

(22) Filed: **Nov. 14, 2005**

(30) **Foreign Application Priority Data**

Jan. 20, 2005 (GB) 0501174.7

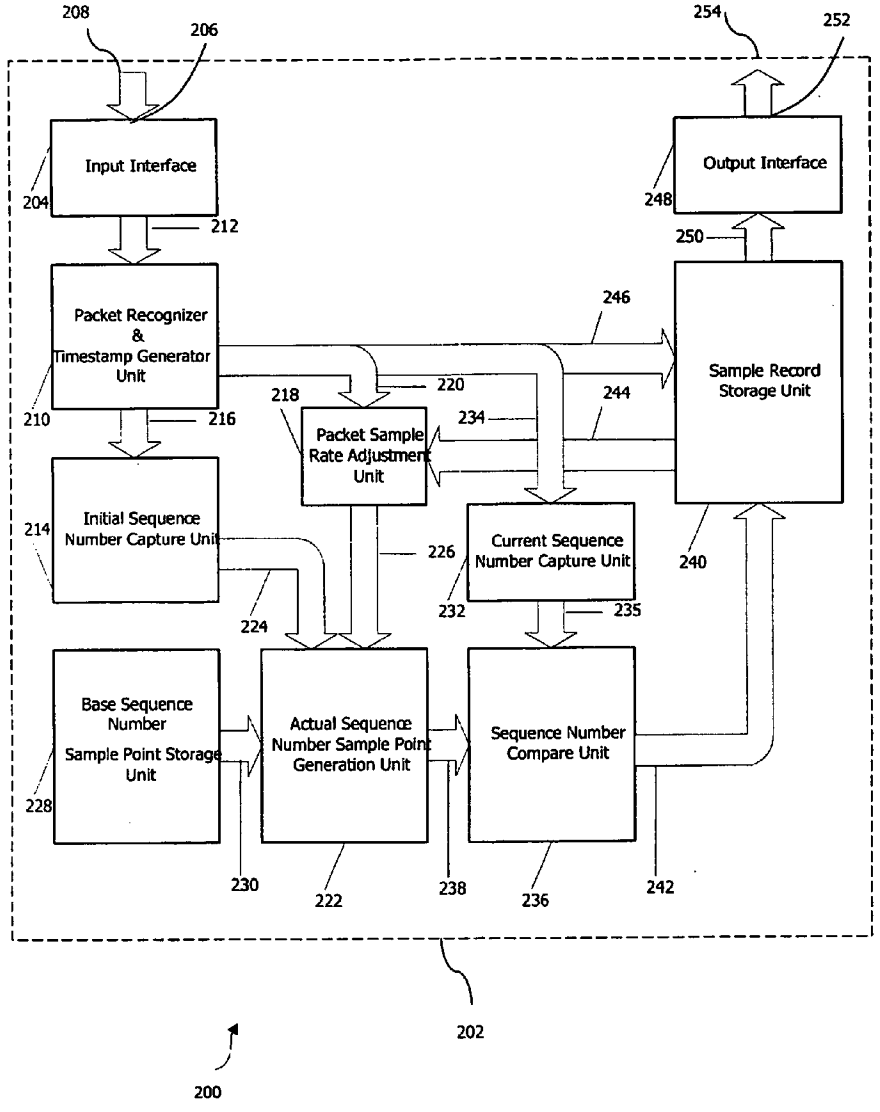
Publication Classification

(51) **Int. Cl.**
H04J 1/16 (2006.01)

(52) **U.S. Cl.** **370/252**; 370/394

(57) **ABSTRACT**

A monitoring system comprises a first network probe coupled to a first point in a communications link and a second network probe coupled to a second point in the communications link. Each of the first and second network probes is arranged to sample packets from a predetermined flow of packets and store timestamps associated with packet samples. The timestamps stored by the probes are subsequently correlated.


208
206
254
252
Input Interface
204
212
Output Interface
248
250
246
Packet Recognizer & Timestamp Generator Unit
210
216
220
218
Packet Sample Rate Adjustment Unit
234
244
Sample Record Storage Unit
Initial Sequence Number Capture Unit
214
224
226
Current Sequence Number Capture Unit
232
235
240
Base Sequence Number Sample Point Storage Unit
228
Actual Sequence Number Sample Point Generation Unit
Sequence Number Compare Unit
230
222
238
236
242
202
200

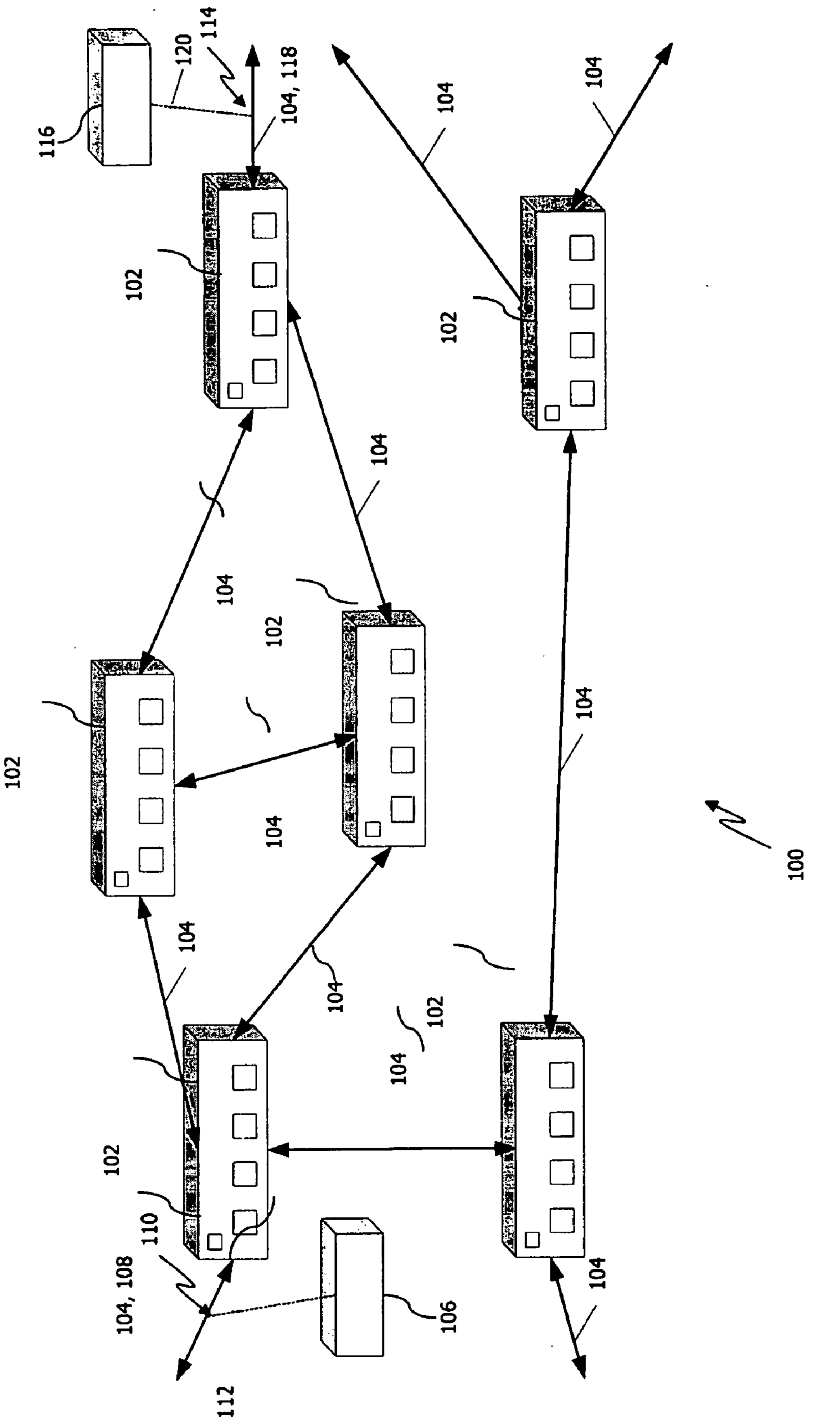

116
120
114
104, 118
102
104
102
104
102
104
102
104
104
102
104
104
102
104
100
104, 108
110
112
106
104

Figure 1

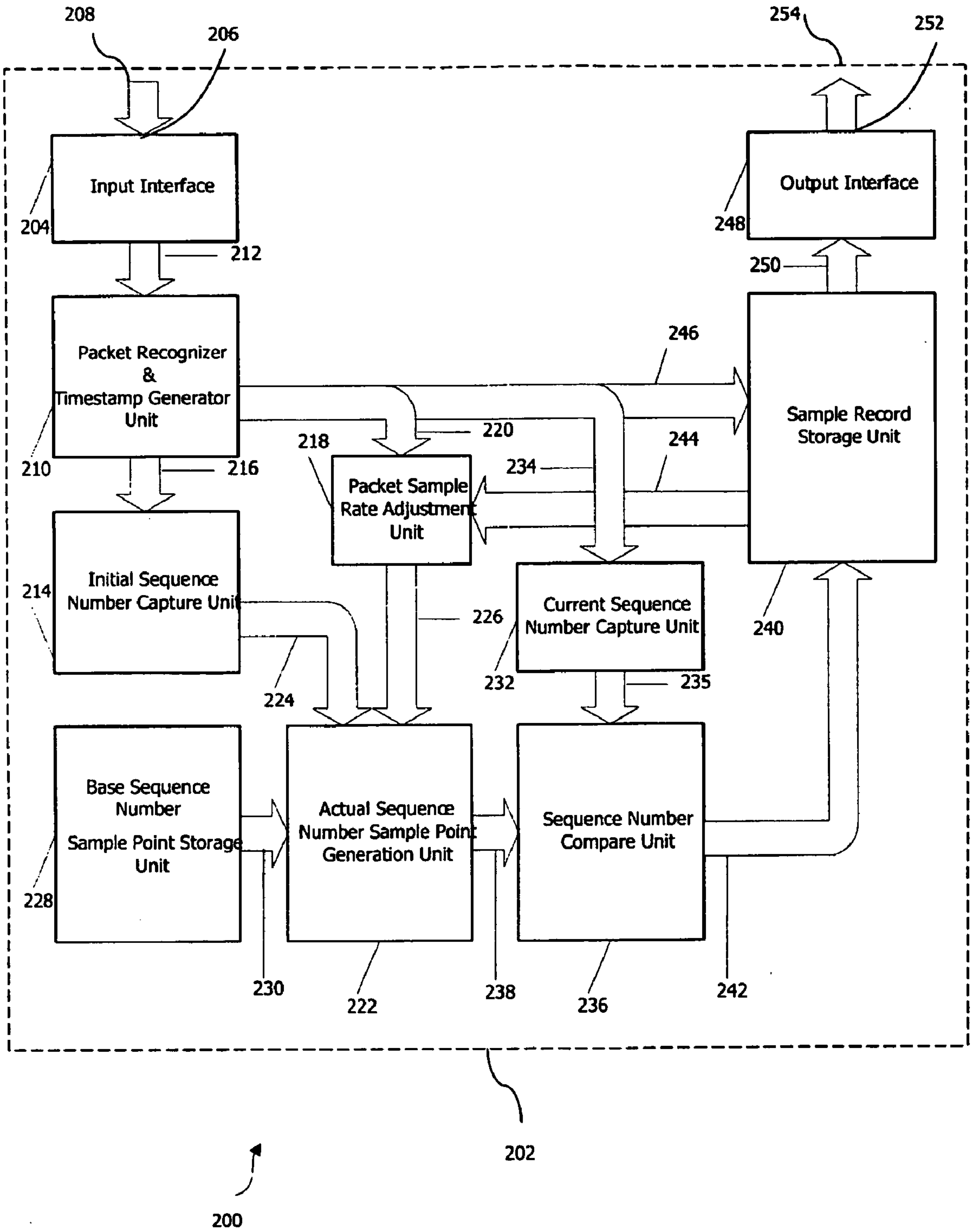

208
206
254
252
Input Interface
204
Output Interface
248
212
250
Packet Recognizer
&
Timestamp Generator
Unit
246
Sample Record
Storage Unit
220
218
244
234
210
216
Packet Sample
Rate Adjustment
Unit
Initial Sequence
Number Capture Unit
214
226
Current Sequence
Number Capture Unit
240
232
235
224
Base Sequence
Number
Sample Point Storage
Unit
Actual Sequence
Number Sample Point
Generation Unit
Sequence Number
Compare Unit
228
230
222
238
236
242
202
200

Figure 2

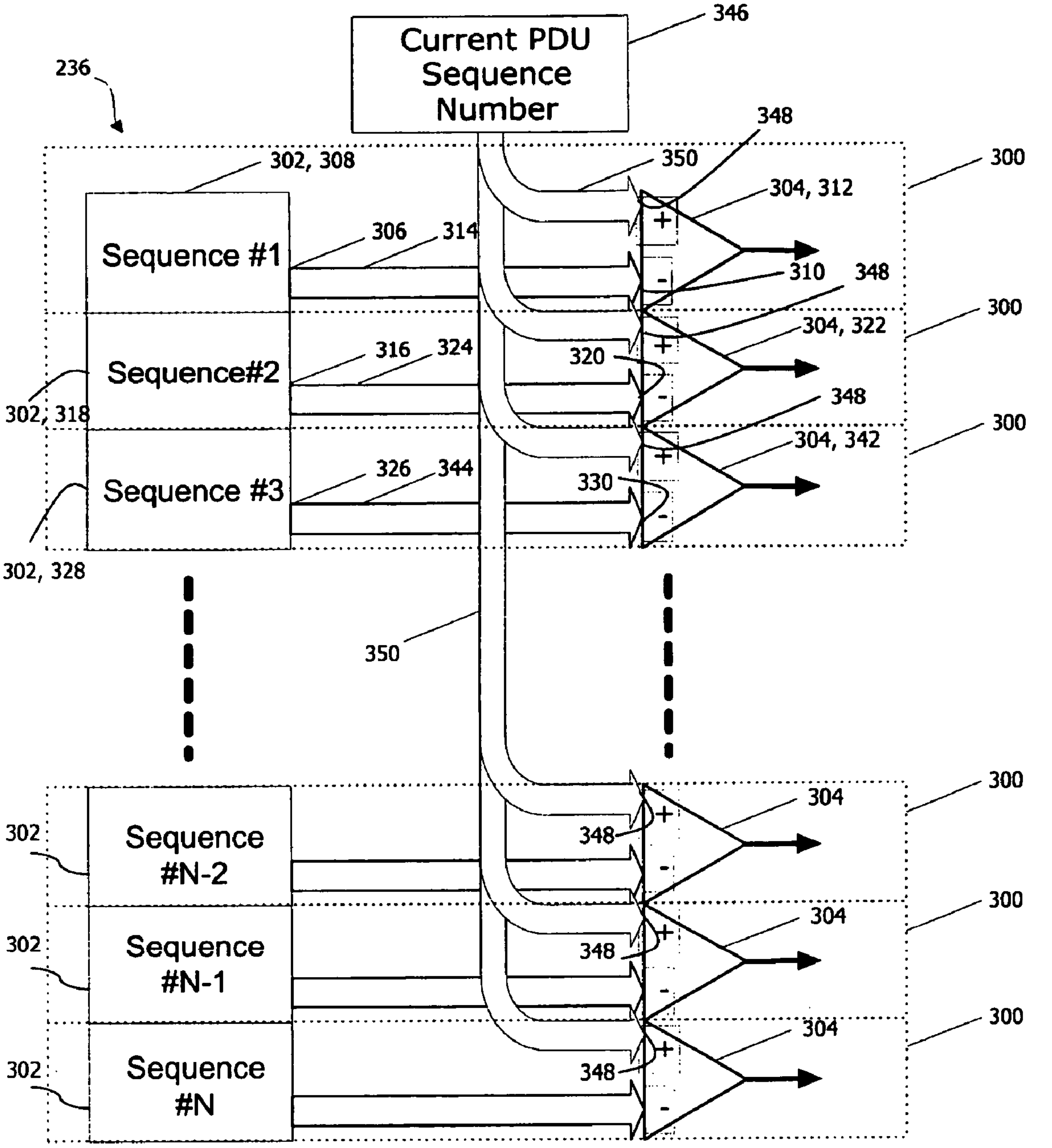

Current PDU Sequence Number
346
236
348
302, 308
350
300
304, 312
306
314
Sequence #1
310
348
300
304, 322
316
324
320
Sequence#2
302, 318
348
304, 342
300
326
344
330
Sequence #3
302, 328
350
300
304
302
Sequence #N-2
348
300
304
302
Sequence #N-1
348
300
304
302
Sequence #N
348
+
-

Figure 3

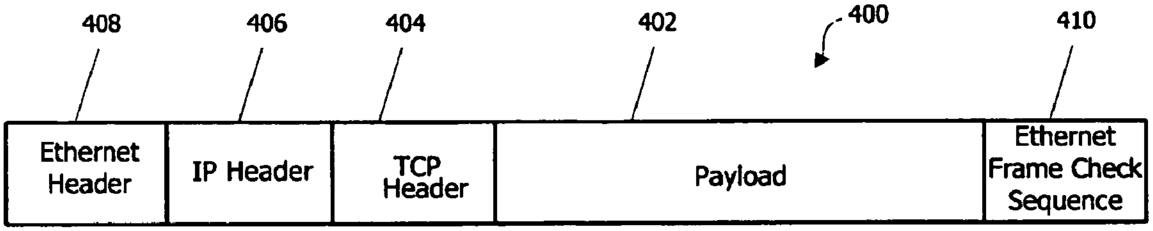


Figure 4

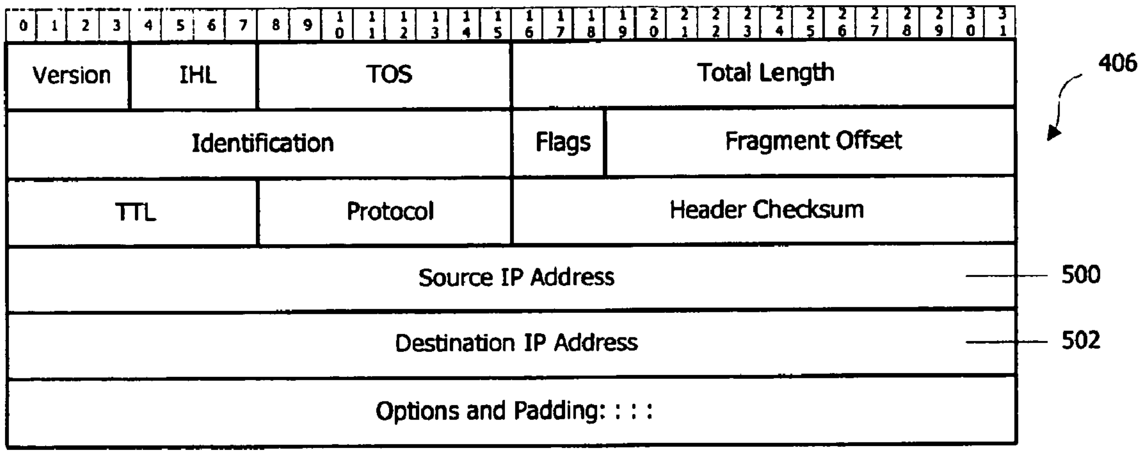


Figure 5

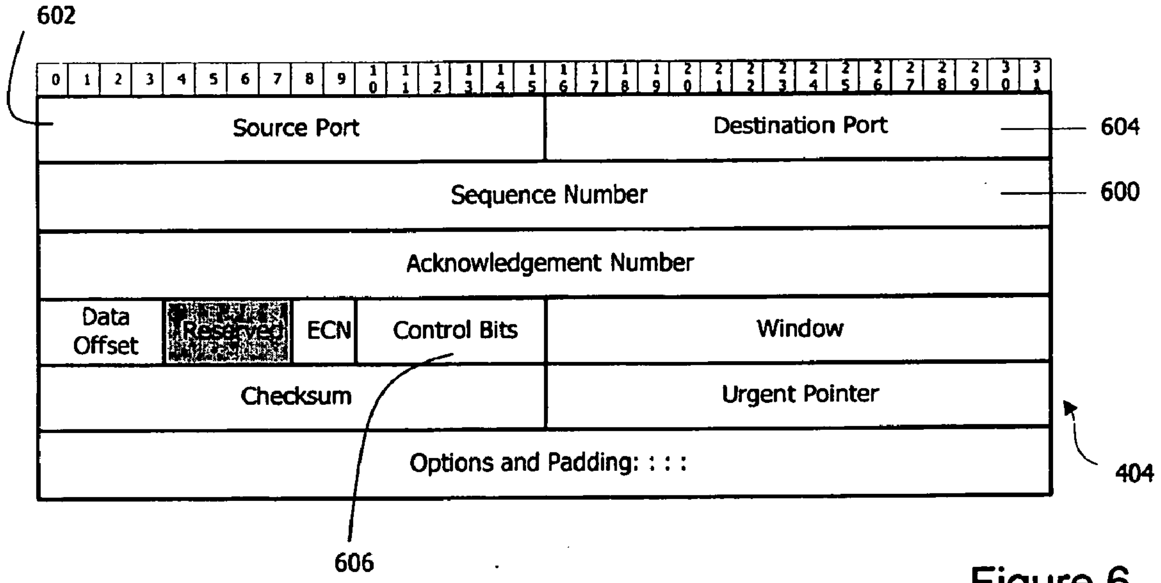


Figure 6

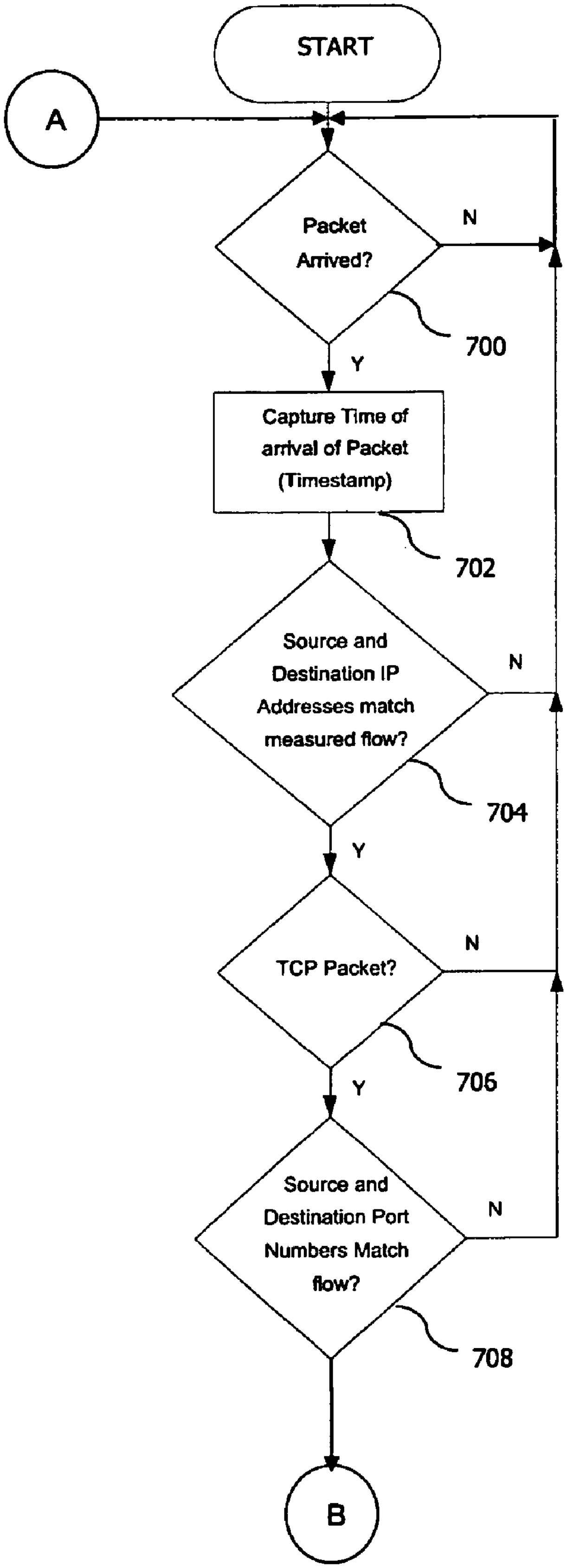

START
A
Packet Arrived?
N
700
Y
Capture Time of arrival of Packet (Timestamp)
702
Source and Destination IP Addresses match measured flow?
N
704
Y
TCP Packet?
N
706
Y
Source and Destination Port Numbers Match flow?
N
708
B

Figure 7A

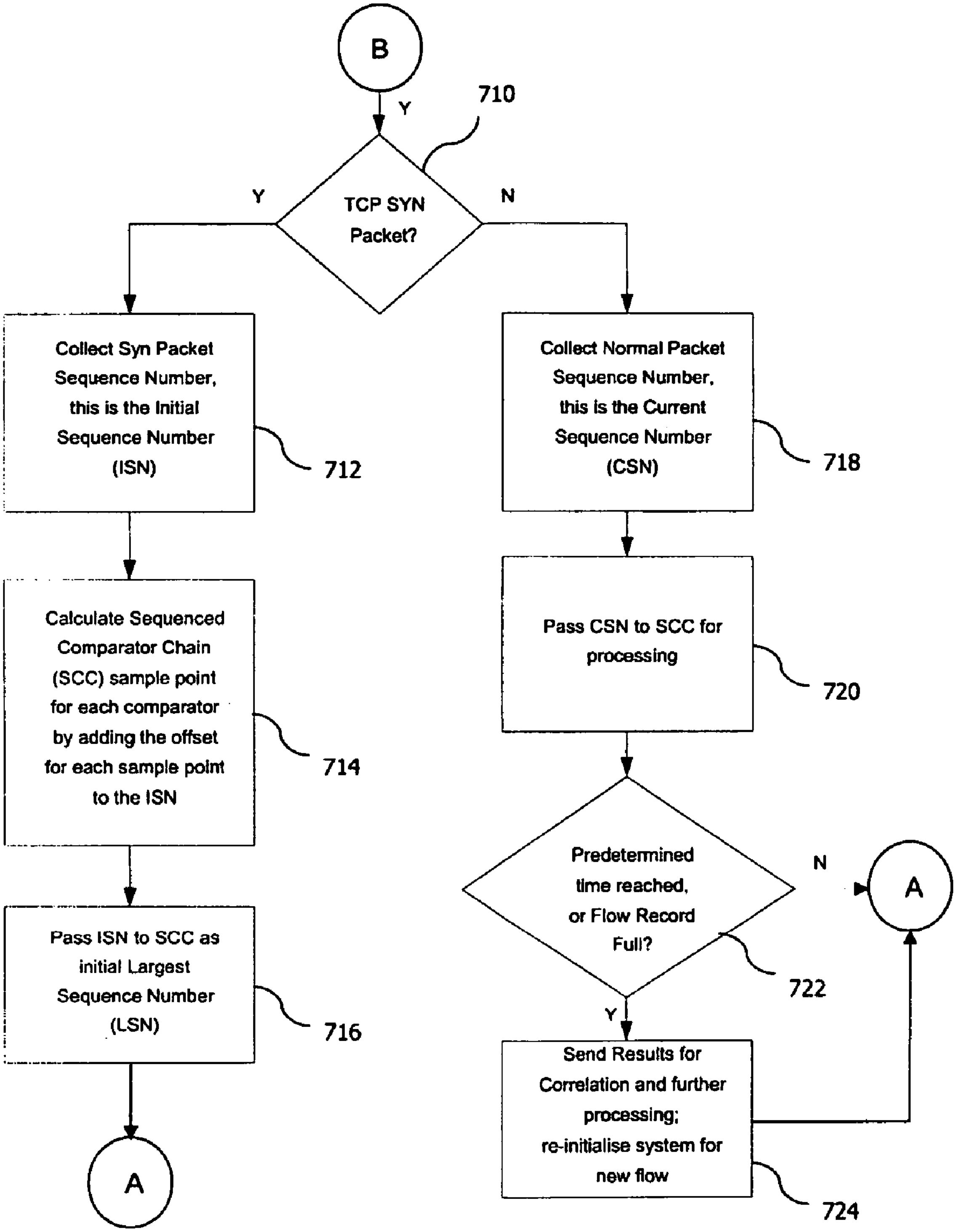

B
Y
710
TCP SYN Packet?
Y
N
Collect Syn Packet Sequence Number, this is the Initial Sequence Number (ISN)
712
Calculate Sequenced Comparator Chain (SCC) sample point for each comparator by adding the offset for each sample point to the ISN
714
Pass ISN to SCC as initial Largest Sequence Number (LSN)
716
A
Collect Normal Packet Sequence Number, this is the Current Sequence Number (CSN)
718
Pass CSN to SCC for processing
720
Predetermined time reached, or Flow Record Full?
722
N
A
Y
Send Results for Correlation and further processing; re-initialise system for new flow
724

Figure 7B

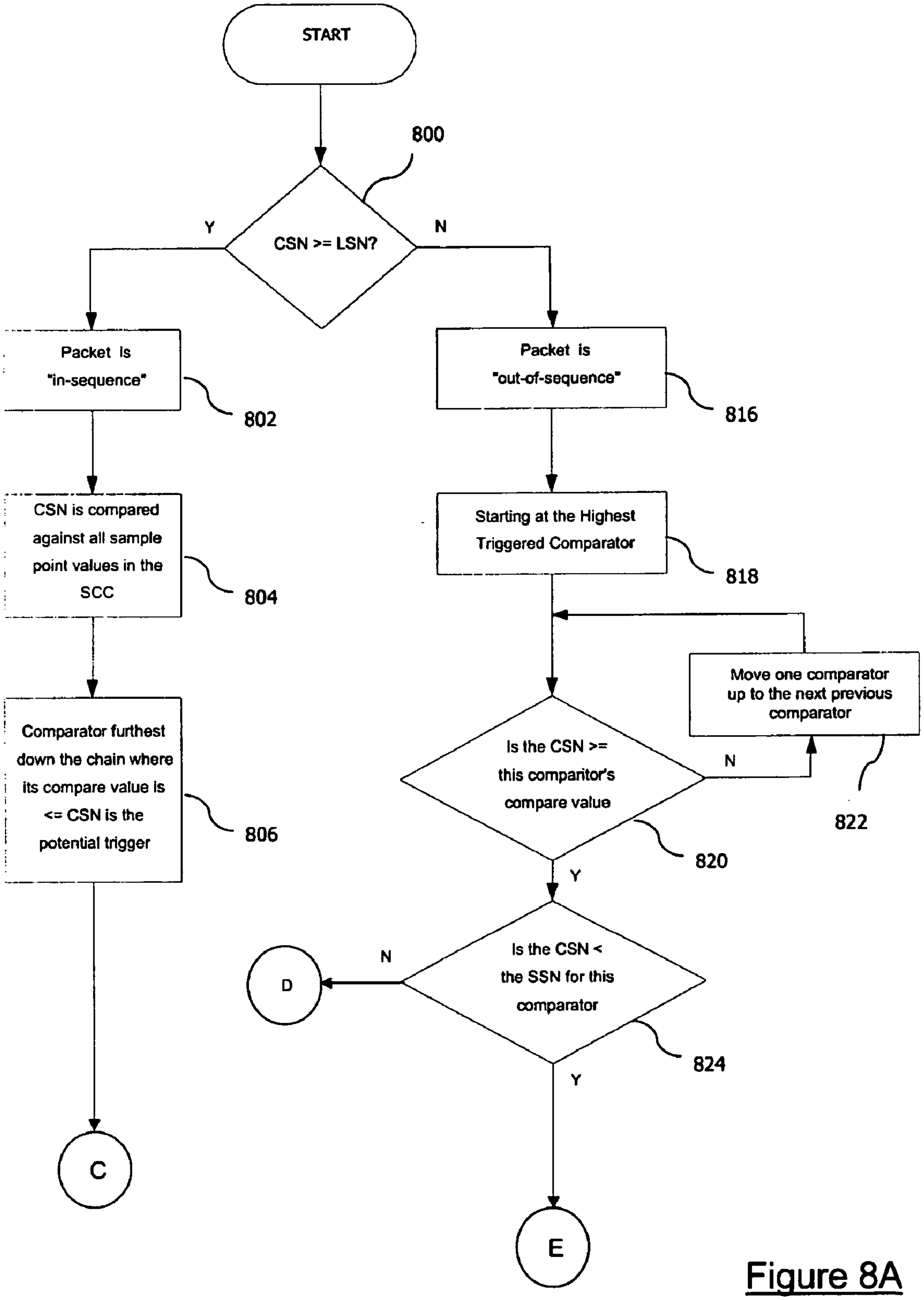

START
800
CSN >= LSN?
Y
N
Packet is "in-sequence"
802
Packet is "out-of-sequence"
816
CSN is compared against all sample point values in the SCC
804
Starting at the Highest Triggered Comparator
818
Move one comparator up to the next previous comparator
822
Comparator furthest down the chain where its compare value is <= CSN is the potential trigger
806
Is the CSN >= this comparitor's compare value
N
820
Y
Is the CSN < the SSN for this comparator
N
D
824
Y
C
E

Figure 8A

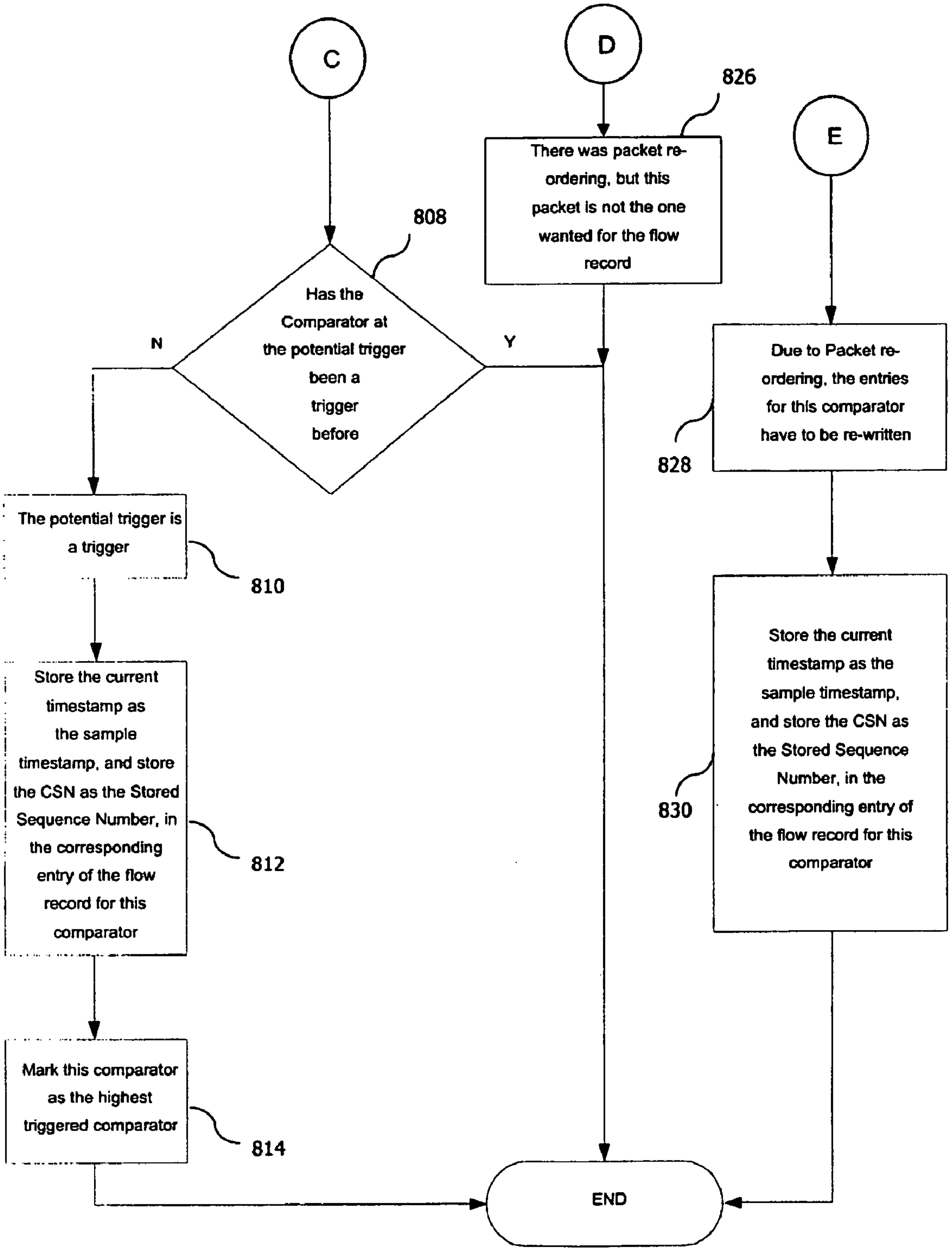

C
D
E
826
There was packet re-ordering, but this packet is not the one wanted for the flow record
808
Has the Comparator at the potential trigger been a trigger before
N
Y
Due to Packet re-ordering, the entries for this comparator have to be re-written
828
The potential trigger is a trigger
810
Store the current timestamp as the sample timestamp, and store the CSN as the Stored Sequence Number, in the corresponding entry of the flow record for this comparator
830
Store the current timestamp as the sample timestamp, and store the CSN as the Stored Sequence Number, in the corresponding entry of the flow record for this comparator
812
Mark this comparator as the highest triggered comparator
814
END

Figure 8B

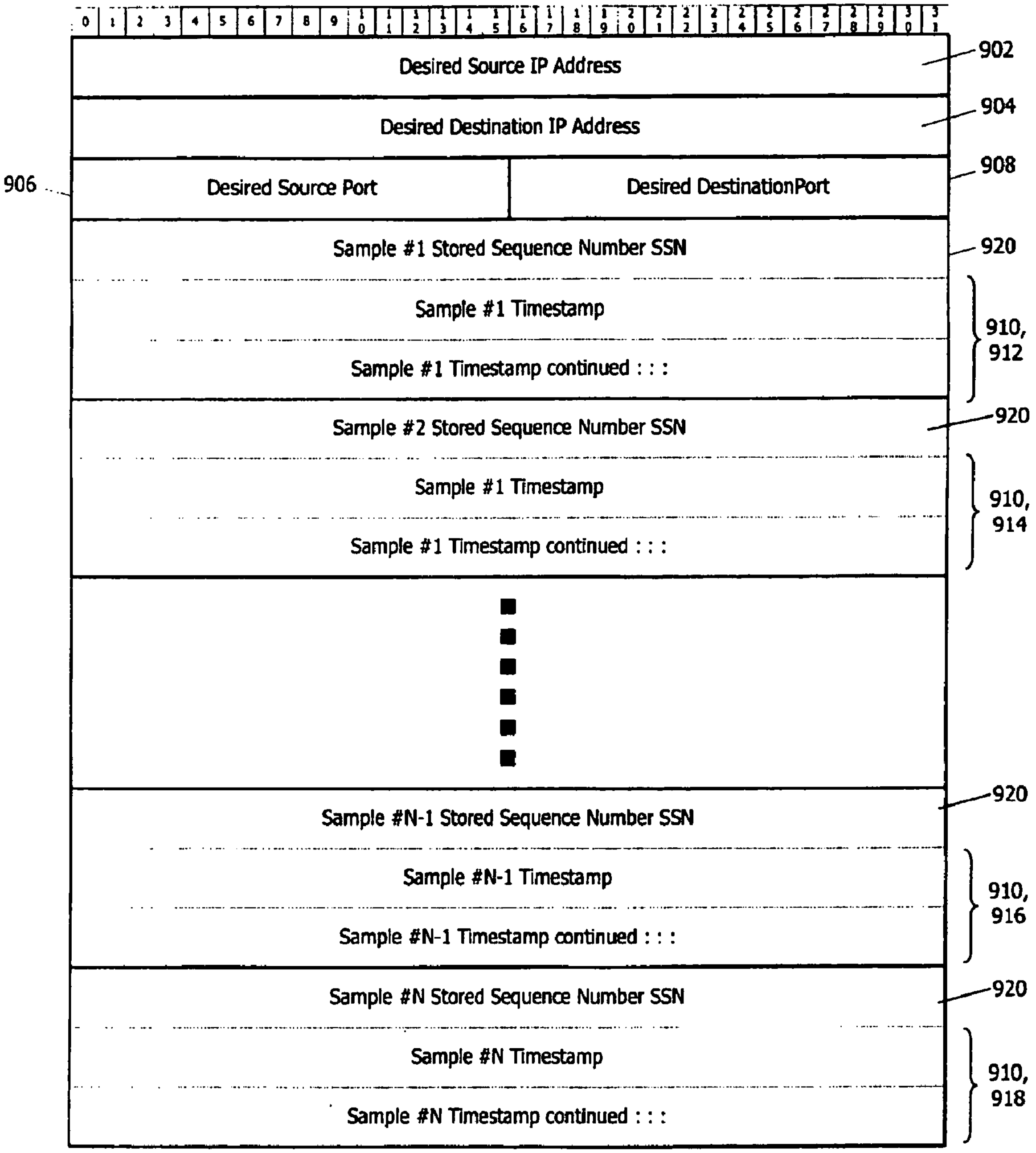

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Desired Source IP Address
902
Desired Destination IP Address
904
906
Desired Source Port
Desired DestinationPort
908
Sample #1 Stored Sequence Number SSN
920
Sample #1 Timestamp
Sample #1 Timestamp continued : : :
910, 912
Sample #2 Stored Sequence Number SSN
920
Sample #1 Timestamp
Sample #1 Timestamp continued : : :
910, 914
Sample #N-1 Stored Sequence Number SSN
920
Sample #N-1 Timestamp
Sample #N-1 Timestamp continued : : :
910, 916
Sample #N Stored Sequence Number SSN
920
Sample #N Timestamp
Sample #N Timestamp continued : : :
910, 918

Figure 9

MONITORING SYSTEM, METHOD OF SAMPLING DATAGRAMS, AND APPARATUS THEREFOR

[0001] The present invention relates to a method of sampling datagrams of the type, for example, that samples a datagram at a first time and at a first point in a communications link and the datagram at a second time and at a second point in the communications link. The present invention also relates to an apparatus for sampling a datagram, for example a network probe. The present invention further relates to a monitoring system of the type, for example, that samples a datagram at a first time and at a first point in a communications link and the datagram at a second time and at a second point in the communications link.

BACKGROUND ART

[0002] In the field of monitoring performance of a communications network, it is known to provide a passive monitoring system to monitor predetermined flows of datagrams communicated over a communications link spanning one or more communications networks. In such monitoring systems, a first probe is located at a first point along the monitored link and a second probe is located at a second point along the monitored link. The probes are coupled respectively to the first and second points along the link by respective taps that "siphon off" a copy of the electrical or optical signals passing the first and second points. The data tapped by the first and second probes are correlated to derive diagnostic data that can be used for managing and troubleshooting the communications network.

[0003] One particular known method that employs the first and second probes as described above, captures a set of packets respectively at the first and second probes, i.e. at the first and second points along the communications link. As each packet is captured, a timestamp is generated as well as a hash signature, derived from the packet data, which uniquely identifies the packet, the hash signature and associated timestamp being stored on the probe. After a predetermined number of packets have been captured, the hash signature and timestamp pairs are sent to a separate processor, known as a correlator, for correlation. At the correlator, the hash signatures are used to identify and match the two observations of a given packet, which passed the two separate points in the communications network. The timestamps of the matched instances of the given packet are then used to calculate jitter and average delay of the set of packets constituting a measured network flow. However, this system may generate sufficient additional traffic that could drive a heavily loaded network into instability.

[0004] Another method of performing loss measurements does not involve forwarding of the hash signature/timestamp pair to a separate correlator. Instead, by filtering at the two probes for a predetermined specific set of matching packet instances corresponding to predetermined flows, only the set of timestamps need be sent for comparison at the correlator. However, this method requires careful selection of a filter algorithm to ensure spurious packets are not selected. If the filter is too wide then not only are too many packets selected, incorrect packets that are part of alternate flows, not part of the measured flow on the monitored link, may also be incorrectly selected for correlation, resulting in spurious timestamps being sent to the correlator and overall results being compromised. Without any mechanism to correlate timestamps with actual packet instances, the above mentioned spurious results cannot be eliminated and a final statistical grooming to remove the spurious results is therefore required. However, even use of complex statistical techniques does not completely eliminate the possibility of pollution of results data by false matches. Additionally, on links with high levels of jitter, the statistical grooming mentioned above may eliminate measurements that are, in fact, correct, thus causing further false results. Additionally, an absence of the hash signature to match timestamps results in the correlator being unable to identify out of order packets.

DISCLOSURE OF INVENTION

[0005] According to a first aspect of the present invention, there is provided a method of sampling datagrams, the method comprising the steps of: sampling a plurality of datagrams from a predetermined flow of datagrams associated with a first point in a communications link, the plurality of datagrams being sampled with reference to a first respective plurality of sampling intervals; generating respective first time record data corresponding to a predetermined number of the plurality of datagrams; sampling the plurality of datagrams from the predetermined flow of datagrams associated with a second point in the communications link, the plurality of datagrams being sampled with reference to a second respective plurality of sampling intervals; generating respective second time record data corresponding to of the predetermined number of the plurality of datagrams; and correlating the first and second time record data; wherein the first respective plurality of sampling intervals is consistent with the second respective plurality of sampling intervals.

[0006] The method may further comprising the steps of: obtaining a copy of datagrams passing the first point and extracting datagrams relating to the predetermined flow of datagrams therefrom; and obtaining a copy of the datagrams passing the second point and extracting datagrams relating to the predetermined flow of datagrams therefrom.

[0007] The sampling may be passive. The first and second respective plurality of sampling intervals may be intervals of datagram sequence numbers.

[0008] The first time record data comprises a first time record entry and the second time record data may comprise a second time record entry, the first time record entry respectively corresponding to the second time record entry, the first and second time record entries relating to a same sampled datagram from the predetermined flow of datagrams.

[0009] The first time record entry may correspond to a first sampling interval of the first respective plurality of sampling intervals, the same sampled datagram having a sequence number numerically closest to a lower limit of the first sampling interval.

[0010] The second time record entry may correspond to a second sampling interval of the second respective plurality of sampling intervals, the same sampled datagram having a sequence number numerically closest to a lower limit of the second sampling interval.

[0011] The step of correlating the first and second time record data may comprise the steps of: receiving the first time record data in respect of the first point, the first time record data comprising a first plurality of time record entries associated with the sampled plurality of datagrams; and receiving the second time record data in respect of the second point, the second time record data comprising a second plurality of time record entries associated with the sampled plurality of datagrams.

[0012] The first time record data and the second time record data may each comprise flow-identifying data to identify the predetermined flow of datagrams.

[0013] The step of sampling the plurality of datagrams from the predetermined flow of datagrams associated with the first point in the communications link may comprise the steps of: providing a plurality of individually testable threshold values, the plurality of threshold values delineating the first respective plurality of sampling intervals; and comparing a first sequence number of a datagram from the predetermined flow of datagrams with each of a first number of the plurality of threshold values so as to identify a first threshold value from the first plurality of threshold values equal to or less than the first sequence number of the datagram and numerically closest to the first sequence number.

[0014] The step of generating the respective first time record data may comprise the step of: recording a first time record in respect of the datagram from the predetermined flow of datagrams in response to the first threshold value identified; wherein the first time record may be recorded in the first time record data, a position of the first time record within the first time record data relative to other time record entries corresponding uniquely to the first threshold value identified from the first number of the plurality of threshold values.

[0015] The step of sampling the plurality of datagrams from the predetermined flow of datagrams associated with the second point in the communications link may comprise the steps of: providing a plurality of individually testable threshold values, the plurality of threshold values delineating the second respective plurality of sampling intervals; and comparing a second sequence number of a datagram from the predetermined flow of datagrams with each of a second number of the plurality of threshold values so as to identify a second threshold value from the second plurality of threshold values equal to or less than the second sequence number of the received datagram and numerically closest to the second sequence number.

[0016] The step of generating the respective second time record data may comprise the step of: recording a second time record in respect of the datagram from the predetermined flow of datagrams in response to the second threshold value identified; wherein the second time record is recorded in the second time record data, a position of the second time record within the second time record data relative to other time record entries corresponding uniquely to the second threshold value identified from the second number of the plurality of threshold values.

[0017] Each time record entry of the first time record data may respectively correspond to each of the first respective plurality of sampling intervals. Each time data entry of the second time record data may respectively correspond to each of the second respective plurality of sampling intervals.

[0018] The first threshold value may be unavailable for subsequent comparisons in response to the first threshold value being equal to or less than the first sequence number of the datagram from the predetermined flow of datagrams and numerically closest to the first sequence number. The second threshold value may be unavailable for subsequent comparisons in response to the second threshold value being equal to or less than the second sequence number of the datagram from the predetermined flow of datagrams and numerically closest to the second sequence number.

[0019] The first and second threshold values may be available for comparison in response to the first and second sequence number being out-of-sequence.

[0020] The predetermined flow of datagrams may comprise an out-of-sequence datagram, the method further comprising the step of: identifying the out-of-sequence datagram as being out of sequence; and identifying one of the first number of the plurality of threshold values less than or equal to a sequence number of the out-of-sequence datagram and numerically closest to the sequence number of the out-of-sequence datagram.

[0021] The method may further comprise the step of: substituting a time record entry for a datagram sampled in the absence of the out-of-sequence datagram with a time record associated with the out-of-sequence datagram.

[0022] The predetermined flow of datagrams may comprise an out-of-sequence datagram, the method further comprising the step of: identifying the out-of-sequence datagram as being out of sequence; and identifying one of the second number of the plurality of threshold values less than or equal to a sequence number of the out-of-sequence datagram and numerically closest to the sequence number of the out-of-sequence datagram.

[0023] The method may further comprise the step of: substituting a time record entry for a datagram sampled in the absence of the out-of-sequence datagram with a time record associated with the out-of-sequence datagram.

[0024] The method may further comprise the step of: temporarily storing sequence numbers of datagrams corresponding to time records stored in the first time record data and/or second time record data; and using the temporarily stored sequence numbers to identify in the first time record data and/or the second time record data the time record entry for the datagram sampled in the absence of the out-of-sequence datagram.

[0025] The step of using the temporarily stored sequence numbers may comprise the steps of: comparing from the temporarily stored sequence numbers a temporarily stored sequence number associated with the time record entry recorded in the absence of the out-of-sequence datagram with the sequence number of the out-of-sequence datagram; and determining whether the sequence number of the out-of-sequence datagram is less than the temporarily stored sequence number.

[0026] The method may further comprise the step of: sending the first time record data and/or the second time record data to a correlator without the temporarily stored sequence numbers.

[0027] The first plurality of threshold values may comprise a predetermined separation therebetween; and the second plurality of threshold values may comprise substantially the predetermined separation therebetween.

[0028] The first plurality of thresholds and/or the second plurality of thresholds may comprise an initial threshold, the method may further comprise the step of: setting the initial threshold with respect to a Transmission Control Protocol (TCP) synchronise (SYN) value.

[0029] The method may further comprise the step of: obtaining the TCP SYN value from a TCP SYN datagram.

[0030] The first and second pluralities of time record entries may be arrival times associated with the sampled plurality of datagrams at the first and second points, respectively.

[0031] According to a second aspect of the present invention, there is provided a computer program element comprising computer program code means to make a computer execute the method as set forth above in relation to the first aspect of the invention.

[0032] The computer program element may be embodied on a computer readable medium.

[0033] According to a third aspect of the present invention, there is provided a method of calculating datagram jitter comprising the method as set forth above in relation to the first aspect of the invention.

[0034] According to a fourth aspect of the present invention, there is provided a method of calculating datagram delay comprising the method of sampling datagrams as set forth above in relation to the first aspect of the invention.

[0035] According to a fifth aspect of the present invention, there is provided a datagram sampling apparatus comprising: a sampler for sampling a plurality of datagrams from a predetermined flow of datagrams associated with a point in a communications link, the plurality of datagrams being sampled with reference to a respective plurality of sampling intervals; and a time record generator for generating respective time record data corresponding to a predetermined number of the plurality of datagrams; wherein the respective plurality of sampling intervals is in accordance with a shared predetermined sampling interval regime so that the time record data comprises a plurality of time record entries corresponding respectively to the respective plurality of sampling intervals.

[0036] The time record data may be contained in a data structure.

[0037] According to a sixth aspect of the present invention, there is provided a time record correlator apparatus for a communications network, the apparatus comprising: a processing resource arranged to receive first time record data and second time record data, and correlate the first and second time record data; wherein the first time record data comprises a first plurality of time record entries and the second time record data comprises a second plurality of time record entries, a position of a data record entry in the first time record data having a corresponding known counterpart position in the second time record data.

[0038] According to a seventh aspect of the present invention, there is provided a datagram sampling system comprising: a first sampler for sampling a plurality of datagrams from a predetermined flow of datagrams associated with a first point in a communications link, the plurality of datagrams being sampled with reference to a first respective plurality of sampling intervals; a first time record generator for generating respective first time record data corresponding to a predetermined number of the plurality of datagrams; a second sampler for sampling the plurality of datagrams from the predetermined flow of datagrams associated with a second point in the communications link, the plurality of datagrams being sampled with reference to a second respective plurality of sampling intervals; a second time record generator for generating respective second time record data corresponding to of the predetermined number of the plurality of datagrams; and a correlator the first and second time record data; wherein the first respective plurality of sampling intervals is consistent with the second respective plurality of sampling intervals.

[0039] It is thus possible to provide a method, system and apparatus that does not require identifying patterns to be sent with timestamps of sampled packets, thereby reducing the amount of data that needs to be communicated to a correlator.

BRIEF DESCRIPTION OF DRAWINGS

[0040] At least one embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

[0041] **FIG. 1** is a schematic diagram of interconnected network elements constituting a part of a communications network and monitoring apparatus coupled thereto;

[0042] **FIG. 2** is a schematic diagram of the monitoring apparatus of **FIG. 1** constituting an embodiment of this invention;

[0043] **FIG. 3** is a schematic diagram of a part of the monitoring apparatus of **FIG. 2**;

[0044] **FIG. 4** is a schematic diagram of a datagram structure used to form messages for the communications network in accordance with the IEEE802.3 standard;

[0045] **FIG. 5** is a schematic diagram of a structure of an Internet Protocol (IP) Header of the datagram structure of **FIG. 4**;

[0046] **FIG. 6** is a schematic diagram of a structure of a Transmission Control Protocol (TCP) Header of the datagram structure of **FIG. 4**;

[0047] **FIGS. 7A and 7B** are flow diagrams of a method of sampling datagrams using, inter alia, the apparatus of **FIG. 2**;

[0048] **FIGS. 8A and 8B** are flow diagrams of a step of **FIG. 7B** in greater detail; and

[0049] **FIG. 9** is a schematic diagram of a data structure for use with the apparatus and/or method of FIGS. **1** to **8**B.

DETAILED DESCRIPTION

[0050] Throughout the following description identical reference numerals will be used to identify like parts.

[0051] Referring to **FIG. 1**, a communications network **100**, for example a part of an intranet, comprises a plurality of interconnected network elements **102**. In this example, the network elements **102** are routers, the network elements **102** being interconnected by respective communications links **104**. A first network probe **106** is coupled to a first link

108 at a first point **110** in the communications network **100** by a first network tap **112**. At a second point **114** in the communications network **100**, a second network probe **116** is coupled to a second link **118** by a second network tap **120**. Although not shown in any detail in **FIG. 1**, the first and second taps **112**, **120** are readily available for most common transport systems; the first and second taps **112**, **120** are therefore appropriately arranged for the networking technology employed within a communications link (described later herein) being monitored and serve to facilitate passive monitoring of the communications link.

[0052] Turning to **FIG. 2**, each of the first and second network probes **106**, **116** comprises a processing resource **200**, for example an Application Specific Integrated Circuit (ASIC). The processing resource **200** provides a packet sampling unit **202** comprising an input interface **204** having a first input **206** coupled to an input data bus **208**. Although not shown in **FIG. 2**, the input data bus **208** is coupled to other electrical circuitry for receiving a stream of bits as an electrical or optical signal via the first or second network taps **112**, **120** and conditioning the received electrical or optical signal prior for receipt by the input interface **204**. The input interface **204** is coupled to a packet recogniser and timestamp generator unit **210** via a first interconnecting data bus **212**. The packet recogniser and timestamp generator unit **210** is coupled to an initial sequence number capture unit **214** via a second interconnecting data bus **216** and to a packet sample rate adjustment unit **218** via a third interconnecting data bus **220**. Both the initial sequence number capture unit **214** and the packet sample rate adjustment unit **218** are respectively coupled to an actual sequence number sample point generator **222** via a fourth interconnecting data bus **224** and a fifth interconnecting data bus **226**. A base sequence number sample point storage unit **228** is also coupled to the actual sequence number sample point generation unit **222** via a sixth interconnecting data bus **230**.

[0053] The packet recogniser and timestamp generator unit **210** is also coupled to a current sequence number capture unit **232** via a seventh interconnecting data bus **234**, the current sequence number capture unit **232** being coupled to a sequence number compare unit **236** via an eighth interconnecting data bus **235**. The sequence number compare unit **236** is also coupled to the actual sequence number sample point generation unit **222** via a ninth interconnecting data bus **238** and to a sample record storage unit **240** via a tenth interconnecting data bus **242**.

[0054] The sample record storage unit **240** is also coupled to the packet sample rate adjustment unit **218** via an eleventh interconnecting data bus **244**, to the packet recogniser and timestamp generator unit **210** via a twelfth interconnecting data bus **246** as well as to an output interface **248** via a thirteenth interconnecting data bus **250**. The output interface **248** has an output **252** coupled to an output data bus **254**. Although not shown in **FIG. 2**, the output data bus **254** is coupled to other electrical circuitry that is arranged to transmit the "results" to a management or measurement device (not shown in **FIG. 1**, but described later herein).

[0055] Although the above units have been described in terms of hardware, the skilled person will, of course, appreciate that the functionality can be implemented in software or a combination of both hardware and software.

[0056] Turning to **FIG. 3**, the sequence number compare unit **236** comprises a chain of N comparator modules **300**. Each comparator module **300** comprises a sequence number register **302**, the sequence number register **302** being coupled to an inverting input of a corresponding comparator **304**. Consequently, a first output **306** of a first sequence number register **308** is coupled to a first inverting input **310** of a first comparator **312** via a first comparator data bus **314**.

[0057] Similarly, a second output **316** of a second sequence number register **318** is coupled to a second inverting input **320** of a second comparator **322** via a second comparator data bus **324**. A third output **326** of a third sequence number register **328** is coupled to a third inverting input **330** of a third comparator **342** via a third comparator data bus **344**. The remaining sequence number registers **302** and comparators **304** of the N comparator modules **300** are similarly configured and so to avoid over-complicating the description of the sequence number compare unit **236**, the remaining comparator modules will not be described further herein.

[0058] A current Packet Data Unit (PDU) or datagram sequence number register **346** is coupled to each non-inverting input **348** of the comparators **304** via a common comparator data bus **350**.

[0059] In operation, packets originating from a source Internet Protocol (IP) address (not shown) pass, in this example, past the first point **110** and the second point **114** on the way to a destination IP address (not shown), the route being taken being considered a communications link. Referring to **FIG. 4**, a typical packet **400** comprises a payload **402** encapsulated by a Transmission Control Protocol (TCP) packet identified as such by a TCP Header **404**. The TCP packet is encapsulated in an IP packet identified as such by an IP Header **406**. In the present example, the IP packet is encapsulated in an Ethernet frame, for example in accordance with the IEEE 802.3 standard, identified as such by an Ethernet Header **408** and an Ethernet frame check sequence **410**.

[0060] In relation to the IP Header **406** (**FIG. 5**), the IP Header **406** comprises a plurality of fields. However, only a Source IP Address field **500** and a Destination IP Address field **502** are of particular relevance to this example and so the other fields will not be described herein, though it should be noted that the skilled person is familiar with the plurality of fields of the IP Header **406**. In relation to the TCP Header **404** (**FIG. 6**), the TCP Header **404** also comprises a plurality of fields. However, only a Sequence Number field **600**, a Source Port field **602**, a Destination Port field **604** and a control bits field **606** are of particular relevance to this example and so the other fields will not be described herein, though it should be noted that the skilled person is familiar with the plurality of fields of the TCP Header **404**.

[0061] Referring to **FIG. 7A**, the packet recogniser and timestamp generator unit **210** awaits (Step **700**) arrival of a packet. Upon receipt, and hence arrival, of a packet, the received packet recogniser and timestamp generator unit **210** captures (Step **702**) a time of arrival of the packet and generates a timestamp associated with the received packet. The packet recogniser and timestamp generator unit **210** is pre-programmed with, or has access to, a list of flows to be monitored. The list of flows comprises at least one identity of at least one predetermined flow of packets. The at least one predetermined flow is identified by a source IP address (not shown) and a destination IP address (not shown) in combination with the source port number (not shown) and the destination port number (not shown). As implied by the above descriptions of the IP Header **406** and the TCP Header **404**, the source and destination IP addresses are obtained from the Source IP Address field **500** and the Destination IP Address field **502**, respectively, of the IP Header **406**, and the source and destination port numbers are obtained from the Source Port field **602** and the Destination Port field **604**, respectively, of the TCP Header **404**. Consequently, in order to determine whether the received packet is part of a predetermined flow listed amongst the at least one predetermined flow, the packet recogniser and timestamp generator unit **210** analyses the IP Header **406** of the received packet to identify (Step **704**) whether the source and destination IP addresses relate to the predetermined flow. If the source address or destination IP address of the received packet does not relate to the predetermined flow, then the packet recogniser and timestamp generator unit **210** discards the packet and associated timestamp and returns to awaiting (Step **700**) receipt of packets. Otherwise, the packet recogniser and timestamp generator unit **210** then determines (Step **706**) whether the IP Header **406** of the received packet encapsulates a TCP packet. If the IP Header **406** of the received packet does not encapsulate the TCP packet, then the packet recogniser and timestamp generator unit **210** again discards the packet and associated timestamp and returns to awaiting (Step **700**) receipt of packets. Otherwise, the source port number and the destination port number found in the Source Port field **602** and the Destination Port field **604**, respectively, are analysed to determine (Step **708**) whether the source and destination port numbers of the received packet relate to the predetermined flow. If the source and destination port numbers do not relate to the predetermined flow of packets, the packet recogniser and timestamp generator unit **210** discards the received packet and associated timestamp and returns to awaiting (Step **700**) receipt of packets.

[0062] Thereafter (**FIG. 7B**), the packet recogniser and timestamp generator unit **210** determines (Step **710**) whether the packet received is a TCP Synchronisation (SYN) packet by reference to the control bits field **606**.

[0063] If the received packet is a TCP SYN packet, the TCP SYN packet is passed to the initial sequence number capture unit **214** and the initial sequence number capture unit **214** extracts (Step **712**) a sequence number of the SYN packet that serves as an Initial Sequence Number (ISN) value and stores the ISN value. The ISN value is then passed (Step **714**) by the initial sequence number capture unit **214** to the actual sequence number sample point generation unit **222**, where the actual sequence number sample point generation unit **222** calculates Sequenced Comparator Chain (SCC) sample points. The SCC sample points are sequence number threshold values calculated using the ISN value and adding increasing offset values to generate each SCC sample point. The offset values are adjustable by the packet sample adjustment unit **218** in the event that the sample record storage unit **240** is filling-up too rapidly, determined by comparison to the number of matching packets identified by the packet recogniser and timestamp generator unit **210**. The sample point generation unit **222** obtains the offset values from the base sequence number sample point storage unit **228**, the base offset values stored by the base sequence number sample point storage unit **228** being, for example, predefined and having a random distribution and being in respect of a flow having an initial sequence number of 0. The offset values are passed from the base sequence number sample point storage unit **228** to the actual sequence number sample point generation unit **222**, where each successive SCC sample point is then generated by adding an increasing offset value to the ISN value:

$$\mathit{SCC}\_\text{sample\_point\_number}(n)=\mathit{ISN}\_\text{value}+\text{offset\_value}(n)$$

Where n=1, 2, 3, . . .

[0064] For example, a first SCC sample point is the ISN value, a second SCC sample point is the sum of the ISN value and a second offset value, a third SCC sample point is the sum of the ISN value and a third offset value, a fourth SCC sample point is the sum of the ISN value and fourth offset value, and so on. However, it should be appreciated that the precise manner in which the offset values are calculated can vary.

[0065] The first, second, third, . . . , nth sample points are then passed to the sequence number compare unit **236** where they are stored in the first, second, third, . . . , nth sequence number registers **302**, respectively. The ISN value is also passed (Step **716**) to the sequence number compare unit **236** and stored as an initial Largest Sequence Number (LSN) value. Thereafter, the packet recogniser and timestamp generator unit **210** continues awaiting (Step **700**) another packet. It will, of course, be appreciated by the skilled person that this process takes place in parallel to the processing described above in relation to the SYN packet.

[0066] If the received packet is determined (Step **710**) not to be a TCP SYN packet, the packet recogniser and timestamp generator unit **210** passes the received packet to the current sequence number capture unit **232**, whereupon the current sequence number capture unit **232** extracts (Step **718**) a sequence number of the received packet and passes (Step **720**) the extracted sequence number to the sequence number compare unit **236**, the sequence number compare unit **236** storing the extracted sequence number in the datagram sequence number register **346** as a Current Sequence Number (CSN) value.

[0067] Referring to **FIG. 8A**, upon receipt of the CSN value, the sequence number compare unit **236** determines (Step **800**) whether the CSN value is greater than or equal to the LSN value in order to determine if the received packet is out-of-sequence. If the CSN value is greater than or equal to the LSN value, the received packet with which the CSN value is associated is deemed to be in-sequence (Step **802**). Thereafter, the CSN value stored in the datagram sequence n umber register **346** is compared (Step **804**) with each SCC sample point stored in the sequence number registers **302** by each of the N comparator modules **300**. As will be appreciated, the comparison operation performed by the N comparator modules **300** results in a number of the N comparator modules **300** generating a logic HIGH output signal constituting "triggering" or activation of each of the number of comparator modules **300**.

[0068] In order to identify the threshold value that is numerically closest to the CSN value, the sequence number compare unit **236** simply selects (Step **806**) a numerically highest of the number of the N comparator modules **300**, i.e. the triggered comparator modules. Thereafter (**FIG. 8B**), the sequence number compare unit **236** determines (Step **808**) if the comparator module **300** having the numerically highest sequence number stored in its sequence number register **302** has been previously selected. To determined previous selection, a flag is assigned to each of the N comparator modules **300** and the flag for the selected comparator module **300** is analysed to determine whether the flag has previously been set.

[**0069**] If, through analysis of the flag associated with the selected comparator module **300**, the selected comparator module **300** is deemed to have been previously selected, the received packet and the associated timestamp are discarded and the packet sampling unit **202** returns to awaiting (Step **700**) receipt of copies of packets. However, if the selected comparator module **300** has been determined not to have been previously selected, the flag associated with the selected comparator module is set to indicate that the comparator module **300** has been selected (Step **810**). The timestamp generated by the packet recogniser and timestamp generator **210** for the received packet is then stored (Step **812**) in a data structure **900** (**FIG. 9**) constituting a record of the predetermined flow. Referring to **FIG. 9**, the data structure **900** is stored in the sample record storage unit **240** and comprises a Desired Source IP Address field **902**, a Desired Destination IP Address field **904**, a Desired Source Port field **906** and a Desired Destination Port field **908**. The Desired Source IP Address field **902**, the Desired Destination IP Address field **904**, the Desired Source Port field **906** and the Desired Destination Port field **908** together identify the predetermined flow being monitored. The data structure **900** also comprises N time record entries **910**, comprising a first time record entry **912**, a second time record entry **914**, . . . , an N-1th time record entry **916** and an Nth time record entry **918**. Each of the time record entries **910** corresponds respectively to each comparator module **300** on a one-to-one basis. Consequently, if the first comparator module is triggered, the timestamp for the received packet is stored as the first time record entry **912**. If the third comparator module is triggered, the timestamp for the received packet is stored as the third time record entry (not shown). Generalising, if the Nth comparator module is triggered, the timestamp for the packet is stored as the Nth time record entry **918**.

[**0070**] In addition to the timestamp for the received packet, the CSN being tested is temporarily stored (Step **812**) in a Stored Sequence Number (SSN) field of the time record entry **910** associated with the comparator module selected. The sequence number compare unit **236** then stores a Highest Triggered Comparator (HTC) identifier to record (Step **814**) the selected comparator module **300** as being a highest selected and triggered comparator module so far. Thereafter (**FIG. 7B**), the packet sampling unit **202** determines (Step **722**) whether a timeout period has expired (not shown in **FIG. 2**) or the data structure **900** is full. If the timeout period has not expired nor the data structure **900** is full, the packet sampling unit **202** returns to awaiting (Step **700**) a next packet. Otherwise, the sampling record storage unit **240** sends the data structure **900** to a central processing resource, for example an Operations Support Systems (OSS) application for correlation. In this example, an OSS correlator application is provided to correlate data structures received from different network probes deployed throughout the communications network **100**. The packet sampling unit **202** then re-initialises itself in readiness to monitor another predetermined flow of packets. In order to monitor a number of flows simultaneously, the packet recogniser and timestamp generator unit **210**, the initial sequence number capture unit **214**, the packet sample rate adjustment unit **218**, the actual sequence number sample point generation unit **222**, the current sequence number capture unit **232**, and the sequence number compare unit **236** could either be instantiated a number of times, or alternatively, be designed to handle different flows of interest concurrently, bearing in mind that only one packet or frame could be received in one direction at a time. However, for the sake of simplicity and clarity of description the multiple instantiations are not shown or described further herein.

[**0071**] The above operation of the packet sampling unit **202** is repeated for subsequently received packets until, as mentioned above, the timeout period expires or the data structure **900** becomes full. Consequently, the data structure **900** is gradually populated with time record entries relating to the predetermined flow of packets. However, in order to fully illustrate operation of the packet sampling unit **202**, it is now assumed that one of the packets received by the packet sampling unit **202** is an out-of-sequence packet that is not a SYN packet, but is from the predetermined flow of packets being monitored. Consequently, the processing in relation to the out-of-sequence packet reaches the stage where the sequence number comparator unit **236** identifies (Step **800**) the out-of-sequence packet as being out-of-sequence, whereafter operation of the packet sampling unit **202** is different to that described above. Clearly, the sequence number comparator unit **236** determines that the CSN value of the out-of-sequence packet is greater than or equal to the LSN value and so the out-of-sequence packet is deemed (Step **816**) out-of-sequence and so the sequence number comparator unit **236** begins a "reverse search" of the already selected comparator modules **300**.

[**0072**] Consequently, the sequence number comparator unit **236** retrieves (Step **818**) the HTC identifier and determines (Step **820**) whether the CSN value of the out-of-sequence packet is greater than or equal to the sequence number stored in the sequence number register **302** of the comparator module **300** associated with the HTC identifer. If the CSN value of the out-of-sequence packet is not greater than or equal to the sequence number stored in the sequence number register **302** associated with the HTC identifier, the sequence number comparator unit **236**, using the flags for each comparator module **300**, identifies (Step **822**) a next numerically highest comparator module **300** that has previously been triggered and selected, and repeats the above comparison of the CSN value with the sequence number stored in the sequence number register **302** of the next numerically highest comparator module **300** that has been triggered and selected. This sub-process of finding next highest triggered and selected comparator modules **300** is repeated until one is found where the CSN value of the out-of-sequence packet is greater than or equal to the sequence number stored in the sequence number register **302** of the triggered and selected comparator module **300** found.

[**0073**] The sequence number comparator unit **236** then accesses the time record entry **910** corresponding to the triggered and previously selected comparator module **300** found and determines (Step **824**) whether the CSN value of the out-of-sequence packet is less than the SSN value stored in the SSN field **920** of the time record entry **910** accessed. If the CSN value of the out-of-sequence packet is not less than the SSN value stored in the SSN field **920** of the time record entry **910** accessed, the sequence number comparator unit **236** concludes (Step **826**) that packet re-ordering has taken place, but that the out-of-sequence packet is not required to be sampled in order to monitor the predetermined flow of packets. Thereafter (**FIG. 7B**), the sequence number compare unit **236** determines (Step **722**) whether the timeout period has expired or the data structure **900** is full as described above.

[**0074**] Alternatively, if the CSN value of the out-of-sequence packet is found to be less than the SSN value stored in the SSN field **920** of the time record entry **910** accessed, the sequence number comparator unit **236** concludes (Step **828**) that the time record entry **910** accessed has to be re-written and so the timestamp associated with the out-of-sequence packet is stored (Step **830**) in place of the time record entry **920** accessed. Additionally, the CSN value of the out-of-sequence packet is stored in the SSN field **920** of the time record entry **910** amended. Thereafter (**FIG. 7B**), the sequence number compare unit **236** determines (Step **722**) whether the timeout period has expired or the data structure **900** is full as described above.

[**0075**] Prior to sending the data structure **900** to the correlator, the sequence number comparator unit **202** removes all SSN fields **920**, and hence entries, from the data structure **900**.

[**0076**] The above described operation is performed by both the first and second network probes **106**, **116** and so for every data structure generated by the first network probe **106**, a counterpart data structure is generated by the second network probe. Moreover, each time record entry of the data structure generated by the first network probe **106** has a counterpart time record entry in the counterpart data structure generated by the second network probe **116**.

[**0077**] At the correlator (not shown), the data structure generated by the first network probe **106** and the data structure generated by the second network probe are received and, using the data in the Desired Source IP Address field **902**, the Desired Destination IP Address field **904**, the Desired Source Port field **906** and the Desired Destination Port field **908**, are identified as relating to the predetermined flow of packets to be monitored. Further, due to the correspondence between the time record entry positions in each of the data structures generated by the first and second network probes **106**, **116**, the correlator is able to match timestamp data for sampled packets and use the matched timestamp data to perform jitter and/or delay calculations in relation to the predetermined flow of packet.

[**0078**] It should be understood that reference herein to the processing resource is intended to embrace either a single data processing entity or a plurality of data processing entities either co-located or distributed.

[**0079**] It should also be appreciated that although the above-described sampling functionality is implemented within a probe, it can alternatively be implemented within a network element or distributed between a probe and a network element, or between a number of network elements.

[**0080**] Although the above examples have been described in the context of packet communication, it should be appreciated that the term "message" is intended to be construed as encompassing packets, datagrams, frames, cells, and protocol data units and so these term should be understood to be interchangeable.

[**0081**] Alternative embodiments of the invention can be implemented as a computer program product for use with a computer system, the computer program product being, for example, a series of computer instructions stored on a tangible data recording medium, such as a diskette, CD-ROM, ROM, or fixed disk, or embodied in a computer data signal, the signal being transmitted over a tangible medium or a wireless medium, for example, microwave or infrared. The series of computer instructions can constitute all or part of the functionality described above, and can also be stored in any memory device, volatile or non-volatile, such as semiconductor, magnetic, optical or other memory device.

1. A method of sampling datagrams, the method comprising the steps of:

sampling a plurality of datagrams from a predetermined flow of datagrams associated with a first point in a communications link, the plurality of datagrams being sampled with reference to a first respective plurality of sampling intervals;

generating respective first time record data corresponding to a predetermined number of the plurality of datagrams;

sampling the plurality of datagrams from the predetermined flow of datagrams associated with a second point in the communications link, the plurality of datagrams being sampled with reference to a second respective plurality of sampling intervals;

generating respective second time record data corresponding to of the predetermined number of the plurality of datagrams; and

correlating the first and second time record data; wherein

the first respective plurality of sampling intervals is consistent with the second respective plurality of sampling intervals.

2. A method as claimed in claim 1, further comprising the steps of:

obtaining a copy of datagrams passing the first point and extracting datagrams relating to the predetermined flow of datagrams therefrom; and

obtaining a copy of the datagrams passing the second point and extracting datagrams relating to the predetermined flow of datagrams therefrom.

3. A method as claimed in claim 1, wherein the first time record data comprises a first time record entry and the second time record data comprises a second time record entry, the first time record entry respectively corresponding to the second time record entry, the first and second time record entries relating to a same sampled datagram from the predetermined flow of datagrams.

4. A method as claimed in claim 1, wherein the step of correlating the first and second time record data comprises the steps of:

receiving the first time record data in respect of the first point, the first time record data comprising a first plurality of time record entries associated with the sampled plurality of datagrams; and

receiving the second time record data in respect of the second point, the second time record data comprising a second plurality of time record entries associated with the sampled plurality of datagrams.

5. A method as claimed in claim 1, wherein the step of sampling the plurality of datagrams from the predetermined flow of datagrams associated with the first point in the communications link comprises the steps of:

providing a plurality of individually testable threshold values, the plurality of threshold values delineating the first respective plurality of sampling intervals; and

comparing a first sequence number of a datagram from the predetermined flow of datagrams with each of a first number of the plurality of threshold values so as to identify a first threshold value from the first plurality of threshold values equal to or less than the first sequence number of the datagram and numerically closest to the first sequence number.

6. A method as claimed in claim 5, wherein the step of generating the respective first time record data comprises the step of:

recording a first time record in respect of the datagram from the predetermined flow of datagrams in response to the first threshold value identified; wherein

the first time record is recorded in the first time record data, a position of the first time record within the first time record data relative to other time record entries corresponding uniquely to the first threshold value identified from the first number of the plurality of threshold values.

7. A method as claimed in claim 1, wherein the step of sampling the plurality of datagrams from the predetermined flow of datagrams associated with the second point in the communications link comprises the steps of:

providing a plurality of individually testable threshold values, the plurality of threshold values delineating the second respective plurality of sampling intervals; and

comparing a second sequence number of a datagram from the predetermined flow of datagrams with each of a second number of the plurality of threshold values so as to identify a second threshold value from the second plurality of threshold values equal to or less than the second sequence number of the received datagram and numerically closest to the second sequence number.

8. A method as claimed in claim 5, wherein the step of generating the respective second time record data comprises the step of:

recording a second time record in respect of the datagram from the predetermined flow of datagrams in response to the second threshold value identified; wherein

the second time record is recorded in the second time record data, a position of the second time record within the second time record data relative to other time record entries corresponding uniquely to the second threshold value identified from the second number of the plurality of threshold values.

9. A method as claimed in claim 1, wherein the predetermined flow of datagrams comprises an out-of-sequence datagram, the method further comprising the step of:

identifying the out-of-sequence datagram as being out of sequence; and

identifying one of the first number of the plurality of threshold values less than or equal to a sequence number of the out-of-sequence datagram and numerically closest to the sequence number of the out-of-sequence datagram.

10. A method as claimed in claim 9, further comprising the step of:

substituting a time record entry for a datagram sampled in the absence of the out-of-sequence datagram with a time record associated with the out-of-sequence datagram.

11. A method as claimed in claim 1, wherein the predetermined flow of datagrams comprises an out-of-sequence datagram, the method further comprising the step of:

identifying the out-of-sequence datagram as being out of sequence; and

identifying one of the second number of the plurality of threshold values less than or equal to a sequence number of the out-of-sequence datagram and numerically closest to the sequence number of the out-of-sequence datagram.

12. A method as claimed in claim 11, further comprising the step of:

substituting a time record entry for a datagram sampled in the absence of the out-of-sequence datagram with a time record associated with the out-of-sequence datagram.

13. A method as claimed in claim 9, further comprising the step of:

temporarily storing sequence numbers of datagrams corresponding to time records stored in the first time record data and/or second time record data; and

using the temporarily stored sequence numbers to identify in the first time record data and/or the second time record data the time record entry for the datagram sampled in the absence of the out-of-sequence datagram.

14. A computer program element comprising computer program code means to make a computer execute the method as claimed in claim 1.

15. A method of calculating datagram jitter comprising the method of sampling datagrams as claimed in claim 1.

16. A method of calculating datagram delay comprising the method of sampling datagrams as claimed in claim 1.

17. A datagram sampling apparatus comprising:

a sampler for sampling a plurality of datagrams from a predetermined flow of datagrams associated with a point in a communications link, the plurality of datagrams being sampled with reference to a respective plurality of sampling intervals; and

a time record generator for generating respective time record data corresponding to a predetermined number of the plurality of datagrams; wherein

the respective plurality of sampling intervals is in accordance with a shared predetermined sampling interval regime so that the time record data comprises a plurality of time record entries corresponding respectively to the respective plurality of sampling intervals.

18. A time record correlator apparatus for a communications network, the apparatus comprising:

a processing resource arranged to receive first time record data and second time record data, and correlate the first and second time record data; wherein

the first time record data comprises a first plurality of time record entries and the second time record data comprises a second plurality of time record entries, a position of a data record entry in the first time record data having a corresponding known counterpart position in the second time record data.

19. A datagram sampling system comprising:

a first sampler for sampling a plurality of datagrams from a predetermined flow of datagrams associated with a first point in a communications link, the plurality of datagrams being sampled with reference to a first respective plurality of sampling intervals;

a first time record generator for generating respective first time record data corresponding to a predetermined number of the plurality of datagrams;

a second sampler for sampling the plurality of datagrams from the predetermined flow of datagrams associated with a second point in the communications link, the plurality of datagrams being sampled with reference to a second respective plurality of sampling intervals;

a second time record generator for generating respective second time record data corresponding to of the predetermined number of the plurality of datagrams; and

a correlator the first and second time record data; wherein

the first respective plurality of sampling intervals is consistent with the second respective plurality of sampling intervals.

* * * * *